



Scope ciclos

Transcripción

[00:00] Entonces ya tenemos una estructura para conseguir imprimir todos los números del 1 al 10. Perfecto. Ahora vamos a poner un poco más difícil este ejercicio. ¿Qué sucedería si yo deseo al final hallar la suma de todos los números del 1 al 10? Por ejemplo si yo quiero encontrar la suma de todos estos números.

[00:28] Vamos a hacer una nueva clase, entonces vamos a crear aquí class, y vamos a poner ejemplo contador 2, para tener una secuencia entre nuestros ejemplos que estamos usando. Perdón, yo aquí me equivoqué, no es contador, es ciclos. Entonces vamos aquí, refactor, rename. Esto es para corregir el nombre de una variable en eclipse.

[00:58] Entonces vamos a ponerle aquí simplemente cambiamos el nombre. Ciclos, finish, y vemos que automáticamente el cambio de nombre aquí adentro también en la declaración de la clase. ¿Esto por qué? Porque tienen que llevar el mismo nombre tanto el archivo como la clase. Bueno. Ese ya es otro tema.

[01:20] Vamos nuevamente a hacer la mala práctica de copiar este código, porque no vamos a avanzar en el mismo para este ejemplo y lo pegamos aquí. Vemos que no está bien formateado. ¿Qué hacemos para eso? Source. Format. Y listo. Ya está todo.

[01:37] Ahora, si nosotros queremos hallar la suma de todos los números del 1 al 10, entonces comenzamos, voy a borrar estos comentarios, podemos

declarar por ejemplo, usar dos variables. Una variable para el contador y otra del tipo entero para ir incrementando el valor en sí de la suma. Cada vez que él pase por esta iteración, vamos sumándole esa cantidad.

[02:07] Entonces, le vamos a decir que esa variable es un tipo entero, vamos a llamarla total, porque es el total que tenemos hasta ese momento, y va a ser igual a cero. Entonces aquí le decimos que total va a ser igual al total más el contador, porque va a ser el número actual que está siendo incrementado, y contador ++ para que él haga la iteración.

[02:35] Guardamos, ejecutamos el código y tenemos el mismo resultado, no hizo absolutamente nada. Esto es porque hemos olvidado imprimir total, entonces vamos a darle aquí, no era contador, era total. Guardamos, ejecutamos y tenemos el mismo resultado. ¿Qué hay de mal en este código? Recordemos nuevamente lo que ya hemos hablado en la clase anterior del scope, del alcance de una variable.

[03:16] En este caso, cada vez que nosotros estamos declarando total, total está inicializando con cero, entonces cada vez que el contador incrementa y él regresa aquí para evaluar la condición y entra nuevamente, total es creado con valor cero, entonces total nunca se va a sumar al contador porque él vale cero y el único valor que vamos a tener es simplemente el valor del contador.

[03:46] ¿Qué podemos hacer en este caso? Declarar total fuera del scope de while. Con eso nosotros ya estamos incrementando la variable total. Ahora, cerramos aquí, este espacio, guardamos, ejecutamos y tenemos aquí claramente la respuesta final que es 55, que es la suma total de todo.

[04:17] Y cada vez que él va iterando, él va imprimiendo un valor, otro valor, otro valor de la suma hasta el momento. ¿Qué sucede si nosotros no queremos todos los valores sino solamente queremos la suma final, el resultado final? Muy fácil. Como total está declarado fuera del scope de while, entonces podemos usar total aquí afuera, aquí. Vamos aquí que está total.

[04:48] ¿Por qué? Porque la ejecución del código sería la siguiente: tenemos total declarado afuera, que está en cero, y por cada vez que esta condición sea verdadera, total se va a incrementar. Total se incrementa y el contador aumenta. Aquí va a dar false en algún momento y el valor de total ya se incrementó x veces.

[05:13] Vamos a probar si esto es cierto. Ejecutamos y en efecto tenemos 55, que sería la suma de todos los números del 1 al 10.